4-2016

Improving Structure MCMC for Bayesian Networks through Markov Blanket Resampling

Chengwei Su
Dartmouth College

Mark E. Borsuk
Dartmouth College

Journal of Machine Learning Research 17 (2016) 1-20 Submitted 8/14; Revised 7/15; Published 4/16

Improving Structure MCMC for Bayesian Networks through Markov Blanket Resampling

Chengwei Su CHENGWEI.SU.TH@DARTMOUTH.EDU
Mark E. Borsuk* MARK.BORSUK@DARTMOUTH.EDU
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755-8000, USA
**Corresponding Author:*
Phone number: 1-603-646-9944
Fax number: 1-603-646-2277

Editor: Max Chickering

Abstract

Algorithms for inferring the structure of Bayesian networks from data have become an increasingly popular method for uncovering the direct and indirect influences among variables in complex systems. A Bayesian approach to structure learning uses posterior probabilities to quantify the strength with which the data and prior knowledge jointly support each possible graph feature. Existing Markov Chain Monte Carlo (MCMC) algorithms for estimating these posterior probabilities are slow in mixing and convergence, especially for large networks. We present a novel Markov blanket resampling (MBR) scheme that intermittently reconstructs the Markov blanket of nodes, thus allowing the sampler to more effectively traverse low-probability regions between local maxima. As we can derive the complementary forward and backward directions of the MBR proposal distribution, the Metropolis-Hastings algorithm can be used to account for any asymmetries in these proposals. Experiments across a range of network sizes show that the MBR scheme outperforms other state-of-the-art algorithms, both in terms of learning performance and convergence rate. In particular, MBR achieves better learning performance than the other algorithms when the number of observations is relatively small and faster convergence when the number of variables in the network is large.

Keywords: probabilistic graphical models, directed acyclic graph, Bayesian inference, Markov chain Monte Carlo

1. Introduction

A Bayesian network (BN) is a compact graphical representation of a multivariate joint probability distribution of variables. A BN is represented in the form of a directed acyclic graph (DAG), with nodes representing variables and directed edges representing probabilistic dependencies. An important feature of a BN is that each variable represented by a node is understood to be conditionally independent of the set of all its predecessors in the DAG, given the states of its parents (Neapolitan, 2004). In other words, the absence of a directly connecting edge between any two nodes implies that the two corresponding variables are independent given the states of the variables represented by intermediate nodes along a

directed path. Thus, direct dependence between variables can be graphically distinguished from simple correlation mediated by other variables.

Algorithms have been developed to attempt to uncover the BN structures consistent with data from complex systems, with the goal of better understanding the direct and indirect mechanisms of action (Daly et al., 2011). This is particularly the case in systems biology, where diverse data on genes, gene expression, environmental exposure, and disease might be used to reveal the mechanistic links between genotype and phenotype (Su et al., 2013). Early attempts to learn BN structures from data focused on identifying the single best-fitting model. However, when the amount of data is relatively small compared to the number of variables, there are likely to be many different structures that fit the available data nearly equally well. The single best-fitting model is often an arbitrary result of the particular states of the variables that happen to have been observed. Further, presentation of this single best model provides no indication of the relative confidence one can have in its particular structural features (i.e., presence or absence of edges).

A Bayesian approach to structure learning allows for a quantification of the strength with which the data and any available prior knowledge jointly support each possible graph structure, in the form of posterior probabilities. Unfortunately, the number of possible DAGs grows super-exponentially with the number of variables being represented, making a full comparison of Bayesian posterior probabilities associated with alternative structures intractable. The Markov Chain Monte Carlo (MCMC) method as applied to graphical structures provides a numerical solution to this problem. In the original version of this algorithm, proposed by Madigan and York (1995), each transition in the Markov Chain consists of essentially a single edge change to the current graph. While this performs relatively well in small domains with 5-15 variables, it is rather slow in mixing and convergence with a larger number of variables. As the size of the structural search space grows, the chain is prone to getting trapped in local high probability regions separated by regions of lower probability regions. As a result, the samples obtained may not be representative of the true posterior distribution.

Friedman and Koller (2003) proposed a variation on the MCMC algorithm that explores the space of topological node orders rather than exact structures. The space of node orders is much smaller than the space of structures and is also likely to be smoother, allowing for better mixing. Thus, order MCMC is generally observed to converge more reliably to the same posterior probability estimates. More recently, Niinimki et al. (2012)introduced an algorithm based on partial, rather than linear, orders to create a still smaller and smoother posterior sampling space. Given a sample of (linear or partial) node orders, it is then straightforward to obtain a sample of graphs consistent with each order. However, as graphs can be consistent with more than one order, the effective prior probability over graphs involves marginalization over orders. This order-modular prior means that it is difficult to explicitly specify priors over graphs (Grzegorczyk and Husmeier, 2008) and that commonly-used order priors, such as the uniform, introduce bias by generating misrepresentative graph priors. Several variants of the order MCMC algorithm have been developed to address this prior bias (Koivisto and Sood, 2004; Eaton and Murphy, 2007; Ellis and Wong, 2008; Niinimki and Koivisto, 2013). Unfortunately, these approaches all incur substantial computational costs, making them impractical for learning the structure of networks with more than about 20 nodes. Recently Masegosa and Moral (2013) proposed a skeleton-based

approach, in which samples are restricted in the DAG space constrained by a given BN skeleton. This method scales to larger networks but performs poorly for small sample sizes.

In the present paper, we seek to improve the poor convergence of the structure MCMC algorithm without substantially increasing the computational costs. One way to do this is to introduce transitions that take larger steps in the structural search space, thus allowing the sampler to more effectively traverse low-probability regions when moving between local maxima and ridges (Koller and Friedman, 2009). With this intent, Grzegorczyk and Husmeier (2008) propose a new edge reversal move which selects an edge from the current MCMC iteration, reverses the direction of this edge, and then resamples new parents for the two nodes at either end of this edge. The idea is to implement a substantial modification to the current DAG that is customized to the new direction of the edge being reversed. The experimental results of Grzegorczyk and Husmeier (2008) indicate that this approach is an improvement over traditional structure MCMC with regard to convergence and mixing and is as computationally efficient as order MCMC.

We see an opportunity to take the approach of Grzegorczyk and Husmeier (2008) a step further by intermittently resampling the Markov blanket of randomly selected nodes. The Markov blanket of a node contains its parents, children, and the other parents of its children. These nodes comprise a module of local relations that shield a node from the rest of the network. These local relations are sometimes too sturdy to modify through small changes, causing the Markov Chain to linger in a region of the structural space for an unnecessarily long time. Our premise is that, by reconstructing the Markov blanket of a node, our search could more readily escape from local maxima, thus substantially improving the mixing of the Markov chain. For ease of comparison, we employ the same notation as Grzegorczyk and Husmeier (2008) and report similar diagnostic results.

2. BN Structure Learning Using MCMC

There are many ways to learn a BN from data. Since the size of search space of BNs is super exponential in the number of variables, it is impractical to do an exhaustive search. Many approaches therefore employ a score-based heuristic search algorithm, which starts with a random or seeded DAG and then proceeds by adding, reversing or deleting one edge at a time to obtain new DAGs. For each DAG, a score is calculated that indicates how well the DAG fits the data. The Bayesian approach to inference specifies that the consistency of a graph with data D is calculated as:

$$P(G \mid D) = \frac{P(D \mid G) \cdot P(G)}{\sum_{G^* \in \Omega} P(D \mid G^*) \cdot P(G^*)}, \tag{1}$$

where $P(G|D)$ is the posterior probability of a particular graph, $P(D|G)$ is the likelihood of the data given that graph and P(G) is the prior probability over the space of all possible graphs ($G \in \Omega$). Given the independence assumptions employed by a BN, and a noninformative graph prior (Grzegorczyk and Husmeier, 2008), the Bayesian score of a DAG, $P(G|D)$, can be factored into a product of local scores for each node X_n given its parents π_n as:

$$P(G \mid D) = \frac{1}{Z_N} \prod_{n=1}^{N} exp(\Psi[(X_n, \pi_n \mid D]), \tag{2}$$

where Z_N is a normalization term, N is the number of nodes, and $\Psi[X_n, \pi_n D]$ are the local scores defined by the probability model chosen for the likelihood $P(D|G)$. The multinomial distribution with a Dirichlet prior is often adopted, as it leads to a closed form expression for the local scores (Cooper and Herskovits, 1992).

The structure MCMC algorithm of Madigan and York (1995) seeks to generate a set of samples of possible structures with relative frequencies that correspond to the Bayesian posterior distribution. These samples can then be used, for example, to estimate the posterior probabilities of particular features of interest (marginalizing over the various structures). Implementations of the structure MCMC algorithm typically employ a Metropolis-Hastings sampler (Giudici and Castelo, 2003). At each step of the Markov chain, a candidate graph $G^{'}$ is proposed that will replace the current graph G with a probability equal to $\mu = min\{1, R(G^{'} \mid G)\}$ where

$$R(G^{'} \mid G) = \frac{P(G^{'} \mid D)}{P(G \mid D)} \cdot \frac{Q(G \mid G^{'})}{Q(G^{'} \mid G)}. \tag{3}$$

$Q(G^{'} \mid G)$ is the proposal probability for a move from G to $G^{'}$ and $Q(G|G')$ is the probability of a reverse move from $G^{'}$ to G. The ratio of these probabilities is referred to as the Hastings factor and serves to correct for any asymmetries that might be present in the proposal moves.

In the conventional structure MCMC algorithm, each step represents a move to a neighboring DAG, which is different in one edge from the current DAG. When Q is a uniform probability over the current structure's neighbors, the Hastings factor is equal to the ratio of number of neighbors for the reverse and forward moves, and the Markov chain is guaranteed to have a stationary distribution equal to the posterior distribution $P(G \mid D)$. Unfortunately, the space of DAGs is super exponential in the number of variables and the posterior landscape is characterized by ridges (representing equally well fitting graphs) separated by valleys. This makes it difficult for the chain to explore the space, making it slow to mix.

Grzegorczyk and Husmeier's (2008) 'new edge reversal'(REV) move replaces the simple single edge addition, removal, or reversal of Madigan and York's algorithm with a more substantial structural change. First, an edge in the current graph G is selected at random for reversal. The nodes at either end of this edge are then orphaned by removing all incoming edges. Next, a new parent set is sampled for each of these edges which contains the reversal of the originally selected edge and does not introduce any directed cycles to the graph. This then becomes the proposed graph $G^{'}$. Grzegorczyk and Husmeier specify the acceptance probability of the REV move, including the appropriate Hastings factor, thus guaranteeing convergence to the correct posterior distribution under ergodicity. As ergodicity is not assured, they intersperse their REV move with traditional structure MCMC moves in their final algorithm

3. Markov Blanket Resampling

Given the success of REV at improving mixing and convergence without added significant computational burden, we believe that there may be value to introducing more extreme steps into the structure MCMC algorithm. In particular, the traditional sampler may rarely leave regions of the structure space that have a similar posterior probability score. Graphs belonging to the same equivalence class, for example, comprise equiprobable ridges.

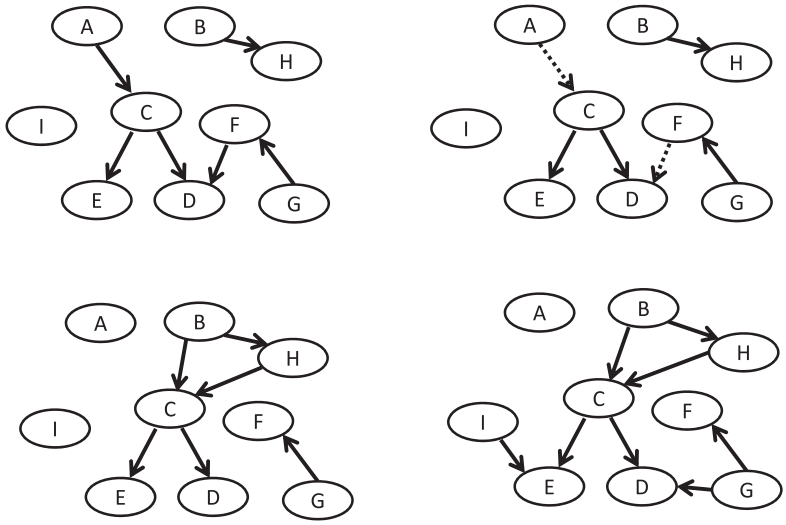

Figure 1: Illustration of the Markov Blanket Resampling (MBR) move. In this example, node C is selected as the target node (top left). Its Markov blanket includes nodes A, D, E, and F. First, all edges pointing to itself and to its children from other nodes are removed (top right). New nodes are next randomly selected as its new parents, in this case nodes B and H (bottom left). Finally, new additional parents are selected for its children, in this case G for D and I for E (bottom right).

Single edge reversals only lead to a new equivalence class when they involve a V-structure. REV improves on this by orphaning two nodes as an intermediate step, but in most cases the chain can easily revert to the same equivalence class within an iteration or two. Thus, even with REV, low probability valleys between ridges may not be readily traversed. As the Markov blanket of a node represents all the variables that can directly influence the state of that node, its entire reconstruction represents the most substantial change that can be made with respect to any single node. Thus, we believe that intermittent resampling of the Markov blanket could further improve the mixing and convergence of a Markov chain.

3.1 Overview

The idea of our Markov Blanket Resampling (MBR) move is shown in Figure 1. We start with the current DAG (upper left). First, a node is selected at random (node C, in this example). All edges pointing into this node are deleted, as well as edges pointing into its children from nodes other than itself (upper right, dotted edges), thus disconnecting much of the node's Markov blanket. New parents for this target node are then sampled disallowing any of its previous parents (bottom left), as are new additional parents, without any restrictions, for all the target node's children (bottom right), both under the condition of not creating any cycles.

3.2 Mathematical Details

Step 1: A node X_i is selected from a DAG, G, according to a uniform distribution over all nodes N in G. Orphaning this node and removing edges pointing into its children from nodes other than X_i creates a new DAG, G_0. (If X_i has no children, then only Step 2 is implemented and the products involving X_j in eq. (9) are set equal to 1.)
Step 2: A new parent set $\pi_i^{'}$ is selected for X_i, disallowing its previous parents π_i, to obtain new DAG, $G_0^{X_i \leftarrow \pi_i^{'}}$. Mathematically, therefore, $\pi_i^{'}$ conforms to the following distribution:

$$\Gamma(\pi_i^{'} \mid G_0, \pi_i) = \frac{exp(\Psi[X_i, \pi_i^{'} \mid D]) \cdot H(G_0^{X_i \leftarrow \pi_i^{'}}) \cdot I(\pi_i^{'}, \pi_i)}{Z^*(X_i \mid G_0, \pi_i)}, \tag{4}$$

where $H(G) = 0$ if the graph G contains a directed cycle and 1 otherwise, $I(\pi_i^{'}, \pi) = 0$ if $\pi \cap \pi_i^{'} \neq \phi$ and 1 otherwise, and $Z^*(X_i|G_0)$ represents the sum of local scores over all allowable parent sets π of X_i

$$Z^*(X_i \mid G_0, \pi_i) := \sum_{\substack{\pi : H(G_0^{X_i \leftarrow \pi}) = 1 \\ \pi_i \cap \pi_i^{'} \neq \phi}} exp(\Psi[X_i, \pi \mid D]). \tag{5}$$

This yields graph G_1.
Step 3: New parent sets $\pi^{'j}_i$ are sampled for each of X_i's children $X_i^j \in \gamma_i = \{X_i^1, ..., X_i^J\}$ that include node X_i and conform to the following distribution:

$$\Gamma(\pi_i^{'} \mid G_j) = \frac{exp(\Psi[X_i^j, \pi^{'j}_i \mid D]) \cdot H(G_j^{X_i^j \leftarrow \pi^{'j}_i}) \cdot J(\pi^{'j}_i, X_i)}{Z(X_i^j \mid G_j, X_i)}, \tag{6}$$

where G_j is the graph that exists prior to sampling new parents for node X_i^j, $J(\pi^{'j}_i, X_i) = 1$ if $X_i \in \pi^{'j}_i$ and 0 otherwise, and $Z(X_i^j|G_j, X_i)$ represents the sum of local scores over all allowable parent sets of X_i^j which contain X_i:

$$Z(X_i^j \mid G_j, X_i) := \sum_{\substack{\pi : H(G_j^{X_i^j \leftarrow \pi}) = 1 \\ X_i \in \pi}} exp(\Psi[X_i^j, \pi \mid D]). \tag{7}$$

It is important to note that sampling of new parent sets for each X_i^j occurs in a specified order that is randomized for each MBR move (see Appendix), so that if another child of X_i (i.e., a 'sibling' of X_i^j) becomes a parent of X_i^j in graph G_{j+1}, then X_i^j is not an allowable parent of that sibling when its new parents are subsequently sampled. This can be considered as a special case of disallowing directed cycles when sampling new parent sets for each X_i^j.

Thus, after these three steps, the DAG $G^{'}$ has been proposed by the MBR move, with overall proposal probability $Q(G^{'} \mid G)$ given by:

$$Q(G^{'} \mid G) = \frac{1}{N} \cdot \frac{exp(\Psi[X_i, \pi_i^{'} \mid D]) \cdot H(G_0^{X_i \leftarrow \pi_i^{'}}) \cdot I(\pi_i^{'}, \pi_i)}{Z^*(X_i \mid G_0, \pi_i)} \cdot \prod_{j=1}^{J} \frac{exp(\Psi[X_i^j, \pi^{'j}{}_i \mid D]) \cdot H(G_j^{X_i^j \leftarrow \pi^{'j}{}_i}) \cdot J(\pi^{'j}{}_i, Xi)}{Z(X_i^j \mid G_j, X_i)}. \tag{8}$$

To compute the acceptance probability of the Metropolis-Hastings algorithm, the probability of performing the complementary reverse move from G' to G, $Q(G \mid G')$, must be articulated. This involves another MBR move for node X_i in which the node is first orphaned and all edges pointing into its children from nodes other than itself are removed to yield DAG, $G_0^{'}$. In step 2, the original parent set π of X_i is selected resulting in graph $G_1^{'}$, while in step 3 the original parent sets of each of X_i's children are selected in graphs $G_j^{'}$ (accounting for node order) thus getting back to the original DAG G.

The numerators of the proposal distributions are cancelled out by the corresponding terms of the posterior ratio (Grzegorczyk and Husmeier 2008), as are all local scores corresponding to unaffected nodes. Therefore, the critical term in the acceptance probability can be written simply as:

$$R(G^{'} \mid G) = \frac{P(G^{'} \mid D)}{P(G \mid D)} \cdot \frac{Q(G \mid G^{'})}{Q(G^{'} \mid G)} = \frac{Z^*(X_i \mid G_0, \pi_i)}{Z^*(X_i \mid G_0^{'}, \pi_i^{'})} \cdot \frac{\prod_{j=1}^{J} Z(X_i^j \mid G_j, X_i)}{\prod_{j=1}^{J} Z(X_i^j \mid G_j^{'}, X_i)}. \tag{9}$$

An outline of the MBR algorithm is given in the Appendix.

3.3 Implementation Details

As with the order MCMC of Friedman and Koller (2003) and REV-enhanced structure MCMC of Grzegorczyk and Husmeier (2008), computational efficiency can be obtained when implementing the MBR algorithm by pre-computing and storing the scores associated with all potential parent sets of each node. Further efficiency can be gained for large networks by only considering as potential parents of each node those which have the highest local scores when considered as sole parents. However, as this is an approximate technique, we do not employ it in the examples presented here. We do, however, employ a fan-in restriction in which a maximum of three parents are allowed for any one node. When we sample for new parents for a node, we test all combinations of at most three. This approximation is used in most applications to reduce computational complexity and improve convergence.

To preserve acyclicity when sampling new parent sets for a node, we can simply eliminate invalid parent sets from the list of potential parent sets. In other words, if a parent set contains a node that is a descendant of the target node or its children, this parent set is removed from the current list of candidate parent sets. This method can also be used to define the scores used in computing the function Z^*. Finally, as the DAGs resulting from the first steps of both the forward and reverse MBR moves, G_0 and $G_0^{'}$, are the same, the identification of descendants only has to occur once during each MBR iteration.

In our applications we implement the MBR move probabilistically in the MCMC algorithm, with a fixed probability of p_m=1/15. The traditional structure move consisting of

the addition, deletion or reversal of a single edge is thus performed with probability p_s=1 p_m. These are the same probabilities as suggested by Grzegorczyk and Husmeier (2008) for their REV move. We experimented with different probabilities but found that 1/15 yields desirable results in terms of prediction quality and convergence rate. More frequent implementations lead to a decline in efficiency in sampling the highest probability regions as the chain seems to jump too often to low probability regions in the structure space, while a lower frequency reduces the mixing of the chain in efficiency incurred by using MBR. Since both the REV and MBR moves scores are pre-computed, MBR does not incur extra running time relative to REV, a fact we confirmed by through time trials using our selected data sets.

4. Experimental Testing

In this section, we test a version of the structure MCMC sampler enhanced by our MBR move against the standard structure sampler and one enhanced by the REV move of Grzegorczyk and Husmeier (2008). We find that the MBR-structure sampler performs better than the other two in terms of reliable convergence and learning performance, especially for relatively small sample sizes. We also find that it converges much faster for large and very large networks. These conclusions are drawn using data simulated from four models, all with more than 30 nodes: ALARM, a BN designed for patient monitoring consisting of 37 nodes and 46 edges (Beinlich et al., 1989); Hailfinder, a BN with 56 nodes and 66 edges developed to forecast severe weather (Abramson et al., 1996) HEPAR II, a BN built for diagnosis of liver disorders consisting of 70 nodes and 123 edges (Onisko, 2003), and GENS2, a tool for simulating genetic epidemiological datasets containing 100 exposure, gene, and disease variables (Pinelli et al., 2012). The ALARM data were used by Grzegorczyk and Husmeier (2008) to demonstrate that the REV-structure sampler performed much better than the standard MCMC sampler and equally well as the order sampler of Friedman and Koller (2003). We employ the other three datasets to compare performance for even larger networks.

Our algorithms were implemented in Matlab on a 128 GB machine with an Intel Xeon 2.00GHz processor. For the ALARM network with a simulated data set of m=1,000, precomputation of scores required 1017.6 sec, and implementation of 10,000 MCMC iterations required 382.6 sec for the standard structure algorithm, 401.5 sec for REV, and 407.3 sec for MBR. Given the 1/15 frequency of implementation, these times imply that the REV and MBR moves require approximately twice as much computation time as the conventional structure moves. This is consistent with the fact that the average number of children per node being considered in each iteration of MBR for ALARM was found to be 1.60.

4.1 Convergence Reliability

To compare the reliability of convergence of the three structure samplers, we generated two MCMC runs using datasets of various sizes (m=50, 100, 250, 500, 1000) simulated from ALARM: one initialized with an empty DAG and another initialized with a DAG identified as highest scoring through an independent run of the MCMC algorithm. We then compared the estimates of the posterior probabilities of the directed edges obtained from the two runs

using scatter plots. Points lying along the 1:1 line imply that the two runs agree, indicating reliable convergence.

We used the same run settings as Grzegorczyk and Husmeier (2008): for the standard structure sampler the burn-in was set to 500,000 iterations and 1000 DAGs were then collected by storing every 1000th iteration. For both the REV and MBR moves, the burn-in length was set to 312,500 and 1000 DAGS were collected by storing every 625th iteration. This difference was employed to ensure approximately equal computational costs across the three samplers, under the conservative assumption that the computational costs of the REV and MBR moves are 10-times greater than those of standard structure iterations and that the probability of these moves in any iteration was 1/15 (Grzegorczyk and Husmeier, 2008).

Scatter plots of the posterior probabilities of directed edges estimated from the two independent runs show that the MBR and REV structure samplers converge approximately equally reliably, even for small sample sizes (Figure 2). There is a high correspondence between the posterior probability estimates for the differently initialized runs. Under the traditional structure sampler, however, the posterior probabilities of some edge features depend on the initialization as indicated by off-diagonal points, even for very large samples, implying that this algorithm tends to get stuck at local maxima. Differences across algorithms also imply imperfect convergence.

4.2 Convergence Rate

Having established that, when added to the traditional MCMC structure sampler, the MBR move improves convergence reliability to the same degree as the REV move, we next compared the convergence rate for networks over a range of sizes: ALARM (37 nodes), HEPAR II (70 nodes), GENS2 (100 nodes). From each network, we generated sets of 1000 simulated observations and applied each of the MCMC algorithms described in the previous section, starting with an empty DAG. Trace plots reveal the speed and reliability of convergence.

For all three networks, the MBR-enhanced algorithm improves the convergence rate substantially over the REV-enhanced version (Figure 3). The highest scoring structures are found in approximately half the number of iterations of REV and in one tenth the number of the traditional structure sampler. In fact, it appears that both the REV and the traditional sampler may be getting stuck at local maxima for the largest networks, while the MBR sampler quickly converges.

4.3 Learning Performance

When the true graph network underlying a dataset is known, we can evaluate the performance of a structure learning algorithm using the concept of AUROC values. If we take the directed edges that have an estimated posterior probability greater than some threshold ϵ to be positive assertions of the existence of that feature, then a comparison against the true network will yield the number of true positive (TP), false positive (FP) and false negative (FN) assertions corresponding to that value of ϵ. The corresponding sensitivity, or true positive rate, TPR=TP/(TP+FN) and the complement of the specificity, or false positive rate, FPR=FP/(TN+FP) can then be calculated, and a plot of these two metrics for varying values of ϵ yields the Receiver Operator Characteristic (ROC) curve. Integrating the ROC

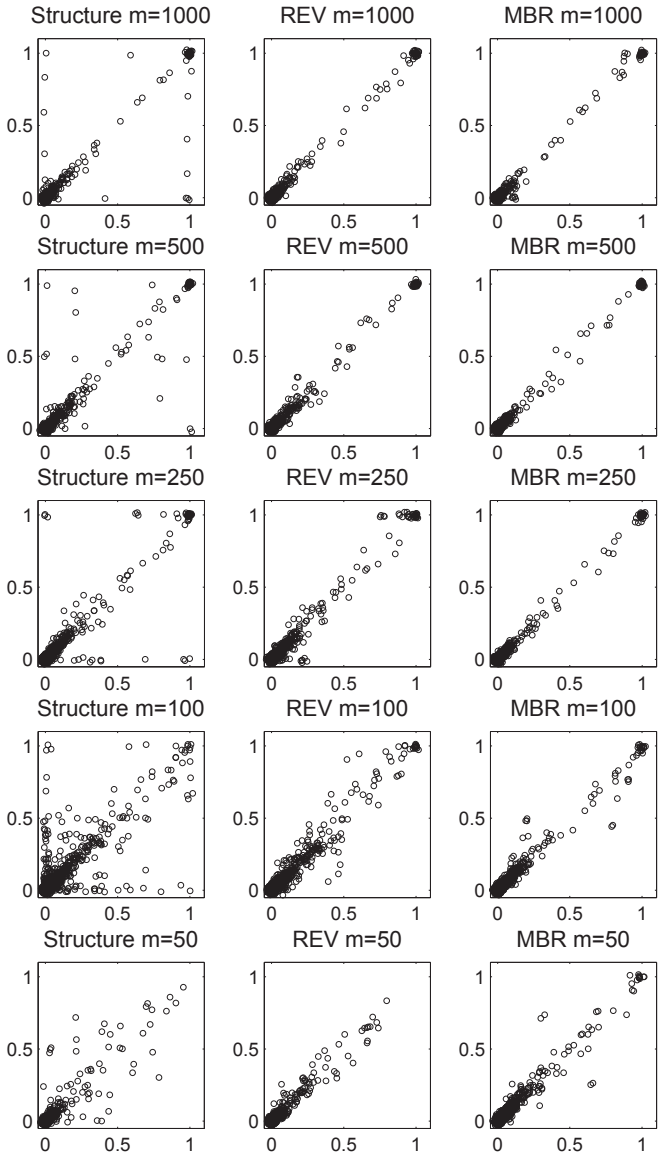

Figure 2: Scatter plots of posterior probability estimates for directed edges in the ALARM network, inferred from simulated datasets of various sizes. In each plot, the x-axis represents the posterior probabilities estimated from an MCMC run initialized with an empty DAG, and the y-axis represents the probability estimates obtained from an MCMC initialized with a high scoring DAG found from an independent run of the MCMC algorithm. When points lie along the diagonal, the two runs have results that agree, indicating reliable convergence. Values clustering near 0 and 1 for larger sample sizes indicate a reduction in inference uncertainty. Left column: traditional structure sampler; center column: REV-enhanced sampler; right column: MBR-enhanced sampler.

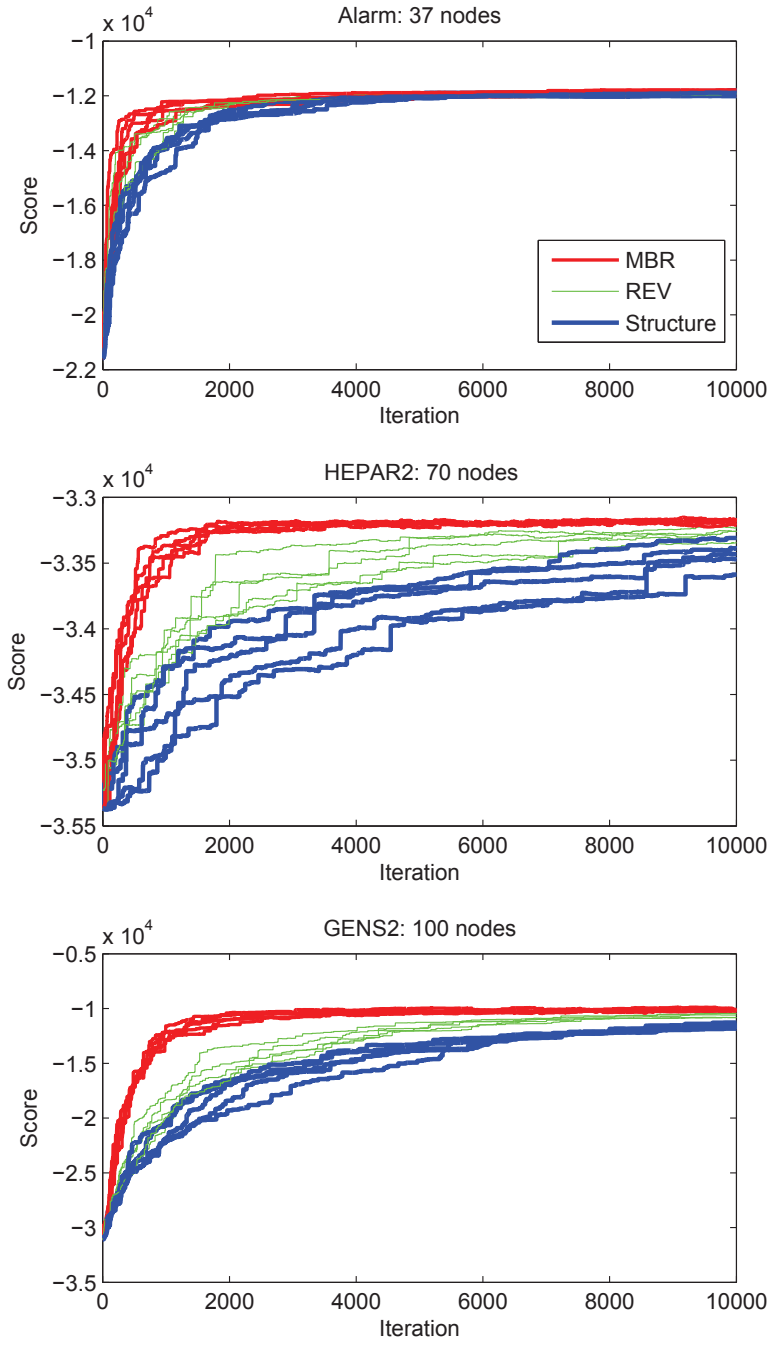

Figure 3: Trace plots of the first 10,000 iterations of five MCMC runs from each algorithm (MBR, REV, Structure) on data sets simulated from networks ranging in size. All runs involved 1000 simulated observations.

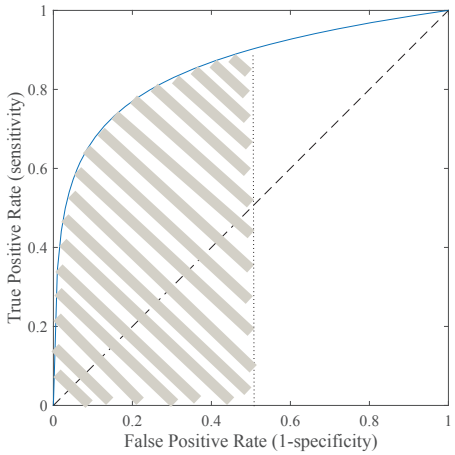

Figure 4: Hypothetical ROC curve. The hatched area represents the $AUROC_{0.5}$ corresponding to an upper limit of ϵ=0.5.

curve from 0 to an upper value of FPR=ϵ then gives the Area under the Receiver Operator Characteristic ($AUROC_\epsilon$) curve as a summary of learning performance, with high values indicating better performance (Figure 4). $AUROC_\epsilon$ values corresponding to low levels of the upper limit are of particular interest, as we are generally concerned with limiting the false positive rate.

For the ALARM network, we calculated $AUROC_\epsilon$ values for graphs learned from datasets of size m = 50, 100, 250, 500, and 1000 observations using each of the three MCMC algorithms and each of the two initializations, as described above. $AUROC_\epsilon$ values were calculated for ϵ = 1.0, 0.1, 0.05, and 0.01. Plots of these values (Figure 5) show that, as expected, learning performance tends to increase with the number of observations. The three algorithms perform nearly comparably at sample sizes of m=250 and greater, with the structure MCMC being somewhat less consistent between initializations, supporting the results shown in Fig. 2. At smaller sample sizes and lower values of ϵ, the traditional structure sampler performs poorly, the REV-enhanced version performs somewhat better, and the MBR-enhanced version performs the best of the three algorithms. Additionally, performance differs very little between the two independent initializations.

To assess learning performance on a different and somewhat larger network, we made comparable calculations with Hailfinder, using simulated datasets of size m = 500, 750, 1000, 1250, and 1500 observations (Figure 6). The traditional structure MCMC again shows increased learning performance with increasing number of observations. Yet, while the REV and MBR-enhanced versions show comparable performance with large datasets, they both yield even better performance with small datasets, as measured by $AUROC_1$, $AUROC_{0.01}$, and $AUROC_{0.05}$ values. This occurs because the Hailfinder network was originally constructed based, in part, on expert judgment. Therefore, the original network is not necessarily the best scoring one. For small sample sizes, the algorithms each find networks that fit the data better (i.e., more concisely) than the original network. Others working with similar-sized networks employing expert judgment or hidden nodes have found similar

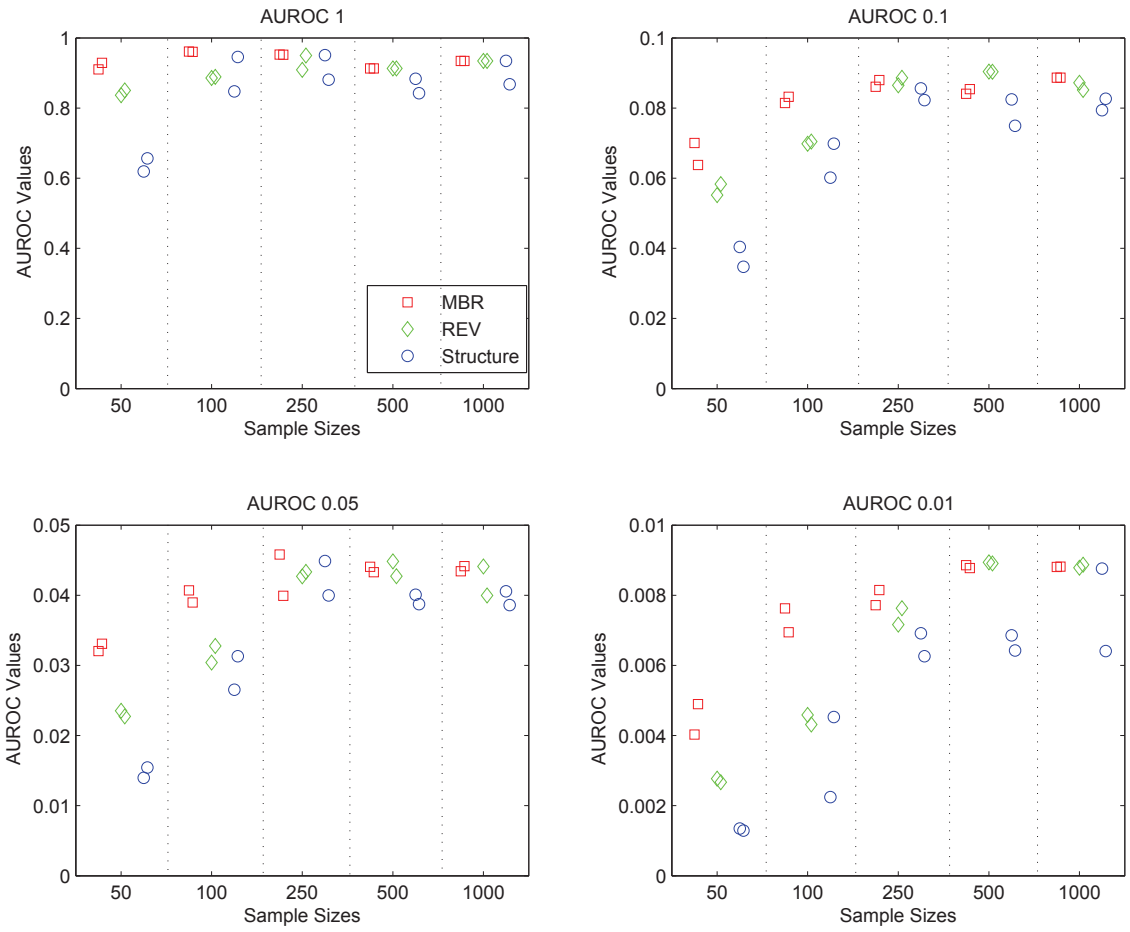

Figure 5: Learning performance of the three MCMC samplers as applied to simulated data from the ALARM network. Each panel shows $AUROC_\epsilon$ values corresponding to a different upper limit on the inverse specificity (ϵ = 1, 0.1, 0.05, 0.01). Within each panel, results are shown for each sampler with a different symbol, as applied to varying numbers of simulated observations (m = 50, 100, 250, 500, 1000) across the x-axis. For each sampler, there are two values, corresponding to runs initialized with an empty DAG (slightly left) and a high scoring DAG found from an independent run of the MCMC algorithm (slightly right).

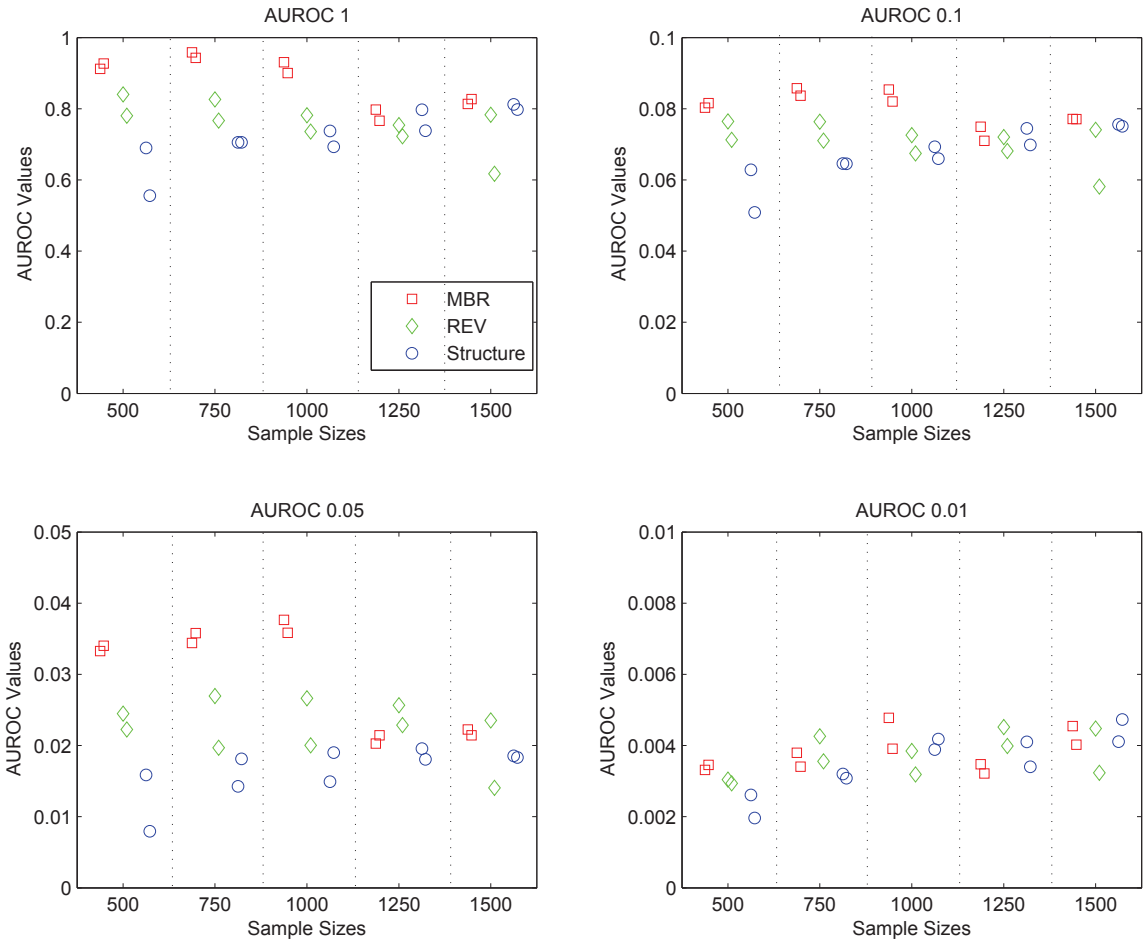

Figure 6: Learning performance of the three MCMC samplers as applied to simulated data from the Hailfinder network. Each panel shows $AUROC_\epsilon$ values corresponding to a different upper limit on the inverse specificity ($\epsilon = 1, 0.1, 0.05, 0.01$). Within each panel, results are shown for each sampler with a different symbol, as applied to varying numbers of simulated observations (m = 50, 100, 250, 500, 1000) across the x-axis. For each sampler, there are two values, corresponding to runs initialized with an empty DAG (slightly left) and a high scoring DAG found from an independent run of the MCMC algorithm (slightly right).

AUROC	1	0.1	0.05	0.01
Test Statistic	3.52	7.71	6.09	7.78
P-value	0.0065	2.96e-05	1.80e-04	2.74e-05

Table 1: Paired t-test results comparing MBR and REV as applied to the Alarm Data

behavior (Masegosa and Moral, 2013). The MBR version appears to perform somewhat better than the REV version at the smallest sample sizes with the possible exception of $\epsilon = 0.01$.

To rigorously compare the learning performance of the three algorithms, we employed a statistical hypothesis testing procedure. First, we produced 10 independent data samples from the Alarm network and generated 10 corresponding MCMC runs for each algorithm starting with an empty structure. $AUROC_\epsilon$ values were calculated from these 10 runs for each algorithm. We then performed paired-sample t-tests in Matlab with null hypothesis that the differences in $AUROC_\epsilon$ values between each pair of algorithms have mean equal to zero (assuming a normal distribution with unknown variance), with a two-sided alternative hypothesis. A paired test was used because we were able to calculate $AUROC_\epsilon$ values for all three algorithms using the same random data sample. Our hypothesis test was implemented for a data sample size of 100.

Results (Figure 7) show that MBR gives consistently greater $AUROC_\epsilon$ values across all four values of and that the within algorithm variance is relatively small compared to the between-algorithm variance, suggesting that the differences are statistically significant. Indeed, results shown in Table 1 indicate that our MBR algorithm statistically outperforms the other two algorithms for the Alarm data of 100 observations. Q-Q plots (not shown) confirm the approximate normality of the $AUROC_\epsilon$ value differences.

5. Conclusions

We have presented a novel MCMC sampling scheme to improve the mixing and convergence of the traditional MCMC algorithm used for probabilistically inferring the structure of BNs from observational data. The idea is to occasionally introduce major moves in the structural search space, thus allowing the sampler to more effectively traverse low-probability regions between local maxima and ridges. Our moves are more extreme than the new edge reversal (REV) move of Grzegorczyk and Husmeier (2008) in that we propose resampling much of the Markov blanket of nodes. As the Markov blanket contains all the variables that can directly influence the state of a node, its resampling represents a substantial change. Yet, as we can derive the complementary forward and backward moves of MBR, the Metropolis-Hastings algorithm can be used to account for any asymmetries that might be present in these proposal moves.

Our experiments across a range of network sizes show that the Markov Blanket Resampling (MBR) scheme outperforms the state-of-the-art new edge reversal (REV) scheme of Grzegorczyk and Husmeier (2008), both in terms of learning performance and convergence rate. In particular, MBR achieves better learning performance than the other algorithms when the number of observations is relatively small and faster convergence when the number

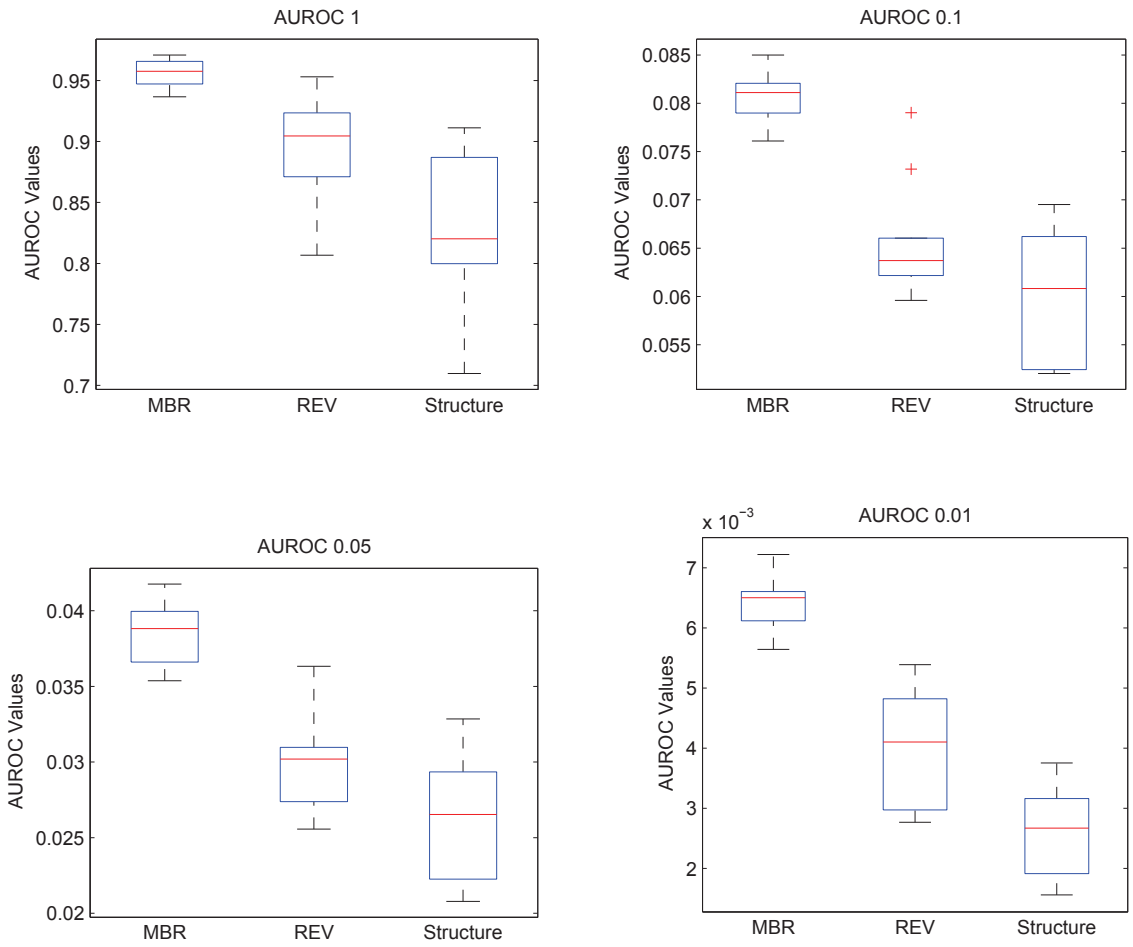

Figure 7: Boxplots of the $AUROC_\epsilon$ values from the three MCMC algorithms as applied to 10 independent sets of simulated data of sample size 100 from the Alarm network. Each panel shows values of $AUROC_\epsilon$ corresponding to a different upper limit on the inverse specificity ($\epsilon = 1,\ 0.1,\ 0.05,\ 0.01$). Boxes indicate the middle 50% (interquartile range, IQR) of the $AUROC_\epsilon$ values for each algorithm, central lines indicate median values, vertical whiskers extend out to the furthest value within 1.5*IQR of the boxes, and crosses indicate outlying values.

of variables in the network is large. As many problems in systems biology are characterized by a large number of variables and a relative paucity of observations (e.g. genome wide association studies), we believe our MBR algorithm will be especially useful in this field. Further, there typically exists prior knowledge (on gene-gene and gene-disease interactions for example) and this prior knowledge can be readily used to improve the performance of the structure-based MCMC algorithms (Imoto et al., 2004; Gao and Wang, 2011; Su et al., 2014), including our MBR scheme. This is not necessarily the case for other algorithms developed to address poor mixing and convergence, such as the order sampler of Friedman and Koller (2003), as discussed in the introduction. Thus, we believe a promising avenue of further research will be to improve methods for constructing informative structure priors from published data, experimental results, and expert opinion.

Acknowledgments

Research supported by grants from the National Center for Research Resources (5P20RR024474-02) and the National Institute of General Medical Sciences (8 P20 GM103534-02) from the National Institutes of Health. We thank the four anonymous reviewers for their thoughtful reviews and constructive suggestions.

Appendix A. Markov Blanket Resampling (MBR) Algorithm

For all domain variables $\{X_1, ..., X_N\}$, pre-compute and store lists of scores of all valid parent sets. We constrain the maximum number of parents a node can have to three in all our tests.

Given the current DAG, perform an MBR move with probability p_m, otherwise perform a traditional single-edge move. If an MBR move is to be performed, then proceed as follows:

- Randomly select a node X_i in current graph G, with probability N^{-1}.
- Store the current parent set π_i of node X_i.
- Delete all edges pointing into X_i and into its children X_i^j with the exception of X_i itself, thus obtaining G_0.
- Find and store the set $D(X_i|G_0)$ of all of X_i's descendant nodes in the current graph G_0.
- In the list of pre-computed scores of possible parent sets of node X_i, mark those that contain a node from the set $D(X_i|G_0)$. Also mark those that contain its parents $_i$ in G_0.
- Sample a new parent set $\pi_i^{'}$ from the unmarked parents sets of X_i according to equation (4). Store the sum of all unmarked scores as the value of $Z^*(X_i|G_0, \pi_i)$.
- Add the new parent set $\pi_i^{'}$ to G_0 to obtain DAG G_1.
- For j from 1 to J (with J being the number of X_i's children), in a specified order that is randomized for each MBR move:

  - Find and store the set $D(X_i^j|G_j)$ of all of X_i^j's descendant nodes in the current graph G_j.
  - In the list of pre-computed scores of possible parent sets of node X_i^j, mark those that contain a node from the set $D(X_i^j|G_j)$. Also mark those that do not contain X_i.
  - Sample a new parent set π'^j_i from the unmarked parents sets of X_i^j according to equation (6).
  - Add the new parent set π'^j_i to G_j to obtain G_{j+1}

- The final graph is the DAG G' being proposed by the MBR move.

- To compute the terms of the complementary inverse move required for calculating the acceptance probability, proceed as follows:

  - Calculate the joint probability of again selecting X_i as N^{-1}.
  - Store the current parent set π_i of node X_i in G'.
  - Delete all edges in G' pointing into X_i and into its children X_j with the exception of X_i itself, thus obtaining G_0'.
  - In the list of pre-computed scores of possible parent sets of X_i, mark those that contain a node in $D(X_i|G_0)$ (which is equivalent to $D(X_i|G_0')$, but has already been computed). Also mark those that contain its parents π'_i in G'. Store the sum of the unmarked scores as the value of $Z^*(X_i|G_0', \pi_i')$.
  - Adding the original parent set π_i to G_0' yields DAG G_1' (which is equivalent to G_1).
  - For j from 1 to J:
    * Find and store the set $D(X_i^j|G_j')$ of all of X_i^j's descendant nodes in the current graph G_j'.
    * In the list of pre-computed scores of possible parent sets of node X_i^j, mark those that contain a node from the set $D(X_i^j|G_j')$. Also mark those that do not contain X_i.
    * Store the sum of the unmarked scores as $Z(X_i^j|G_j', X_i)$.
    * Adding the original parent set π_i^j to G_j' yields G'_{j+1}.

- Calculate the overall acceptance probability of the move from G to G' according to equation (9) using the stored values.

- If the move is accepted replace the current DAG G with G', otherwise keep the current DAG.

References

B. Abramson, J. Brown, W. Edwards, A. Murphy, and R. L. Winkler. A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The Alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. *Lecture Notes in Medical Informatics*, 38:247–256, 1989.

G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

R. Daly, Q. Shen, and S.Aitken. Learning Bayesian networks: Approaches and issues. *The Knowledge Engineering Review*, 26(2):99–157, 2011.

D. Eaton and K. Murphy. Bayesian structure learning using dynamic programming and MCMC. *Proceedings of the Twenty-Third Conference on Uncertainty in Artifical Intelligence*, 2007.

B. Ellis and W. H. Wong. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008.

N. Friedman and D. Koller. Being Bayesian about network structure. a Bayesian approach to structure discovery in bayesian networks. *Journal of the American Statistical Association*, 50(1-2):95–125, 2003.

S. Gao and X. Wang. Quantitative utilization of prior biological knowledge in the Bayesian network modeling of gene expression data. *BMC Bioinformatics*, 12(1):359, 2011.

P. Giudici and R. Castelo. Improving Markov Chain Monte Carlo model search for data mining. *Machine Learning*, 50(1-2):127–158, 2003.

M. Grzegorczyk and D. Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265–305, 2008.

S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology*, 2(1):77–98, 2004.

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

D. Madigan and J. York. Bayesian graphical models for discrete data. international statistical review. *Revue Internationale de Statistique*, pages 215–232, 1995.

A. R. Masegosa and S. Moral. New skeleton-based approaches for Bayesian structure learning of Bayesian networks. *Applied Soft Computing*, 13(2):1110–1120, 2013.

R. E. Neapolitan. *Learning Bayesian Networks*. Pearson Prentice Hall Upper Saddle River, 2004.

T. Niinimki and M. Koivisto. Annealed importance sampling for structure learning in Bayesian networks. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

T. Niinimki, P. Parviainen, and M. Koivisto. Partial order MCMC for structure discovery in Bayesian networks. *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2012.

A. Onisko. *Probabilistic Causal Models in Medicine: Application to Diagnosis in Liver Disorders*. 2003.

M. Pinelli, G. Scala, R. Amato, S. Cocozza, and G. Miele. Simulating gene-gene and gene-environment interactions in complex diseases: Gene-Environment Interaction Simulator 2. *BMC Bioinformatics*, 13(1):132, 2012.

C. Su, A. Andrew, M. R. Karagas, and M. E. Borsuk. Using Bayesian networks to discover relations between genes, environment, and disease. *BioData Mining*, 6(1):6, 2013.

C. Su, A. Andrew, M. R. Karagas, and M. E. Borsuk. Incorporating prior expert knowledge in learning Bayesian networks from genetic epidemiological data. *Computational Intelligence in Bioinformatics and Computational Biology*, 2014.